Cost Sensitive Classification in Data Mining

Zhenxing Qin, Chengqi Zhang, Tao Wang, and Shichao Zhang

Faculty of Information Technology, University of Technology, Sydney
PO Box 123, Broadway, Sydney, NSW 2007, Australia
{zqin,chengqi,wangtao,zhangsc}@it.uts.edu.au

Abstract. Cost-sensitive classification is one of mainstream research topics in data mining and machine learning that induces models from data with unbalance class distributions and impacts by quantifying and tackling the unbalance. Rooted in diagnosis data analysis applications, there are great many techniques developed for cost-sensitive learning. They are mainly focused on minimizing the total cost of misclassification costs, test costs, or other types of cost, or a combination among these costs. This paper introduces the up-to-date prevailing cost-sensitive learning methods and presents some research topics by outlining our two new results: lazy-learning and semi-learning strategies for cost-sensitive classifiers.

Keywords: Cost sensitive learning, misclassification cost, test cost.

1 Introduction

Standard classification is a technique of training classifiers from a given dataset and predicting a query with the trained classifiers. The principal objective of standard classification is a high overall classification accuracy, or the lowest overall misclassification rate. This must lead to an inherent bias in favor of the majority classes because the rare (or minority) class has less impact on accuracy. Although standard classification has successfully been used in many real applications, it was found to, however, fail to meet the need of medical diagnosis data analysis. This is because standard classification is based on the assumption that the class distribution is not very skewed and all classification errors involve the same cost. However, in most data mining and machine learning applications, different misclassification errors often involve different costs.

For example, in medical diagnosis domain, there are 1,479,350 US people have been confirmed as cancer patients. Comparing to the US population, 300 millions, it is rare (about 0.48%). If you classify all people as non-cancer, you could get 99.5% accuracy. However, it is obvious not a solution for our application because the class distribution and impacts are hugely different here. On the other word, misclassifying a Cancer patient as normal could lead his/her death while misclassifying a normal patient as with Cancer may only cost more time/money for further examinations.

We can easily find similar situations in terrorist detection, home loan or credit applications where some classes are rare but with much great impact. Traditional data mining methods that aimed at minimizing error rate will perform poorly in these

L. Cao, J. Zhong, and Y. Feng (Eds.): ADMA 2010, Part I, LNCS 6440, pp. 1–11, 2010.

areas, as they assume equal misclassification cost and relatively balanced class distributions. This leads to the development of domain-driven learning techniques, referred to *cost-sensitive learning*, aiming to address classification problems with non-uniform costs.

Cost-sensitive learning is a procedure of inducing models from data with unbalance class distributions and impacts by quantifying and tackling the unbalance. It has attracted extensive attentions and become one of active research topics in data mining and machine learning, since the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning (ICML-2000). Consequently, diverse learning methods were reported on minimizing the total cost of misclassification costs, test costs, and other types of cost. To make the grasp of these techniques and algorithms easier and smarter, in this paper we survey the up-to-date prevailing cost-sensitive learning methods and classify them into five current research directions. This survey is focused on some basic cost-sensitive learning concepts such as cost function and cost matrix, class probability estimate, misclassification costs and other types of costs involved in the learning process.

2 Settings and Definitions of Cost-Sensitive Learning

2.1 Types of Cost

According to Turney's paper [1], there are nine major types of cost involved in cost-sensitive learning. Two of most studied costs are listed below:

Misclassification Costs: In data mining, different types of misclassification errors usually involve different costs. These costs are the most important costs in cost-sensitive learning. They can either be stationary (represented as a cost matrix) or example dependent.

Test Costs: In some domains, such as medical diagnosis, many tests involve costs. Some tests are more costly than other. If the misclassification costs surpass the test costs greatly, then all tests should be performed. If the test costs are much more than the misclassification costs, then it is rationale not to do any tests.

In addition to the above costs, there are other types of costs such as teacher costs, computation costs, intervention costs, unwanted achievement costs, human-computer interaction costs, costs of cases and costs of instability. In this survey, we concentrate on the cost-sensitive learning methods which minimize the misclassification costs and the test costs.

2.2 Cost Matrix and Cost Function

Most of the cost-sensitive learning methods surveyed in this paper assume that for an M-class problem, an M by M cost matrix C is available at learning time. The value of $C(i, j)$ is the cost involved when a test case is predicted to be class i but actually it belongs to class j.

In reality, $C(i, j)$ can be example dependent, can be represented by a function, but in this survey, most cost-sensitive learning methods assume that $C(i, j)$ does not change during the learning or decision making process. So the cost matrix C is static.

A static cost matrix always has the following structure when there are only two classes:

Table 1. Two-Class Cost Matrix

	Actual negative	Actual positive
Predict negative	C(0, 0) = C00	C(0, 1) = C01
Predict positive	C(1, 0) = C10	C(1, 1) = C11

As per above cost matrix, the cost of a false positive is C_{10} while the cost of a false negative is C_{01}. Conceptually, the cost of labelling an example incorrectly should always be greater than the cost of labelling it correctly. Mathematically, it should always be the case that $C_{10} > C_{00}$ and $C_{01} > C_{11}$ [3].

As we just mentioned, the cost values of a static cost matrix are always fixed, usually defined by dollar values associated with the correct or incorrect decisions. The learning task is not altered if all the cost values are scaled by a constant factor.

Cost values can represent either costs or benefits, or both, and a careful analysis is needed prior to designing the matrix so that all potential costs that are incurred by a decision are captured. Costs are represented by positive values, whereas benefits are represented by negative values [4].

As per Elkan [3], if a cost matrix C is known in advance, let the (i, j) entry in C be the cost of predicting class i when the true class is j. If $i = j$ then the prediction is correct, while if the prediction is incorrect. The optimal prediction for an example x is the class i that minimizes:

$$L(x,i) = \sum_{j=1}^{n} p(j \mid x) C(i,j) \quad . \tag{1}$$

In this framework, a test example should always be predicted to have the class that leads to the lowest expected cost, where the expectation is computed using the conditional probability of each class given the example. The role of a learning algorithm is to produce a classifier that for any example can estimate the probability $P(j|x)$ of each class j being the true class of x. For an example x, making the prediction i means acting as if i is the true class of x. The essence of cost-sensitive decision-making is that it can be optimal to act as if one class is true even when some other class is more probable.

2.3 Traditional Cost-Sensitive Learning Methods

Cost-sensitive learning is an extension of traditional non-cost-sensitive data mining. Cost-sensitive learning methods are also developed based on the existing non-cost-sensitive data mining methods. To make an error-based classifier cost-sensitive, a common method is to introduce biases into an error based classification system in the following different ways [12]:

1) By changing the class distribution of the training data, including:

- Re-sampling
- Instance weighting
- Metacost

2) By modifying the learning algorithms, including:

- Modifying Decision Tree algorithm
- Modifying Naïve Bayes algorithm
- Modifying Neural Network algorithm
- Modifying Support Vector Machine algorithm

3) By taking the boosting approaches, including:

- AdaBoost / AdaCost
- Cost boosting
- Asymmetric boosting

4) Direct cost-sensitive learning which uses the conditional probability estimates provided by error based classifiers to directly compute the optimal class label for each test example using cost function, including:

- Laplace correction
- Smoothing
- Curtailment
- Binning NB
- Platt Calibration
- Isotonic Regression

5) Other cost-sensitive learning methods such as:

- Cost-sensitive specification
- Cost-sensitive CBR system
- Cost-sensitive genetic programming

Among these methods, the "Changing the class distribution of the training data" approach incorporates the misclassification cost into the data pre-processing step. It does this by re-sampling or re-weighting the training data in proportion of their misclassification cost. While the "Direct cost-sensitive learning" approach incorporates the misclassification cost into the data post-processing step. It uses the probability estimation generated by error based classifiers and the cost function to directly compute the optimal class labels for each test example. This approach is easy to implement, but needs good calibration methods to generate accurate probability estimation. The "Modifying the learning algorithms" approach is more straightforward, it modifies the error based classifiers directly to handle misclassification cost, but each classifier needs to be modified separately. The "Boosting" approach is more complicated compared to other approaches. It generates a set of different weak classifiers in sequential trial and then constructs a composite classifier by voting them. The advantage of this approach is that it is applicable to any kind of error based classifiers.

3 Improvement Efforts for Cost-Sensitive Learning

3.1 Cost-Sensitive Learning with Test Costs

The cost-sensitive learning methods mentioned in last section mainly focus on reducing the misclassification cost. Recently, researchers started to consider both test cost and misclassification cost [1, 2, 8-10]. The objective is to minimize the expected total costs of test and misclassification.

Test cost is the cost for obtaining the attribute's value. Current test cost sensitive learning framework combines both of the misclassification cost and test cost together. It aims to minimize the sum of the two kinds of costs. When we combine the test cost and misclassification cost into the classification framework, we need to add extra test cost in the formula defined in Section 2.2, i.e. total cost of all tested attributes for making a decision. Assume there are *m* attributes for test and each attribute *k* is with a test cost t_k, the cost set is noted as T= {t_1, $t_{2,}$ ... t_m}. The optimal prediction for an example x in test cost sensitive learning is class *i* that minimizes:

$$L'(x,i) = \sum_{j=1}^{n} p(j \mid y) C(i,j) + \sum_{k=1}^{m} p(k) * t_k \quad . \tag{2}$$

Where L(x, i) is defined as the formula in Section 2.2; *p*(k) is the probability of performing a test for the value attribute k while making the decision.

Test cost-sensitive learning is an extension of classic cost-sensitive learning. When all the test costs are set as zero, the objective of test cost-sensitive learning is the same as that of classic cost-sensitive learning. Classic cost-sensitive learning framework is a special case of test cost-sensitive learning framework.

Turney [2] developed a learning system, called ICET, a cost-sensitive algorithm that employs genetic search to tune parameters used to construct decision trees. Each decision tree is built using Nunez' ICF criterion (described at the end of this section), which selects attributes greedily, based on their information gain and costs. Turney's method adjusts the test costs to change the behavior of Nunez' heuristic so that it builds different trees. These trees are evaluated on an internal holdout data set using the real test costs and misclassification costs. After several trials, the best set of test costs found by the genetic search is used by the Nunez' heuristic to build the final decision tree on the entire training data set. Because Turney simply modifies the attribute selection in C4.5 to add attribute costs when implementing the Nunez' criterion, his algorithm can deal with continuous attributes and with missing attribute values. Turney [2] is also a seminal work laying the foundations of cost-sensitive learning with both attribute costs and misclassification costs. Turney compares his algorithm with C4.5 and with algorithms sensitive only to attribute costs [5-7]. He does not compare ICET with algorithms sensitive to misclassification costs only, because in his experiments he used simple misclassification cost matrices (equal costs on diagonal, equal costs off diagonal) which make algorithms sensitive only to misclassification costs equivalent to minimizing 0/1 loss. ICET outperformed the simpler greedy algorithms on several medical domains from the UCI repository.

In [8], the cost-sensitive learning problem is cast as a Markov Decision Process (MDP), and an optimal solution is given as a search in a state space for optimal policies. For a given new case, depending on the values obtained so far, the optimal policy can suggest a best action to perform in order to both minimize the misclassification and the test costs. While related to other work, their research adopts an optimal search strategy, which may incur very high computational cost to conduct the search.

Similar in the interest in constructing an optimal learner, Greiner et al. [9] studied the theoretical aspects of active learning with test costs using a PAC learning framework. It is a theoretical work on a dynamic programming algorithm (value iteration) searching for best diagnostic policies measuring at most a constant number of attributes. Their theoretical bound is not applicable in practice, because it requires a specified amount of training data in order to obtain close-to-optimal policies.

Ling et al. [10] proposed a new method for building and testing decision trees involving misclassification cost and test cost. The task is to minimize the expected total cost of tests and misclassifications. It assumes a static cost structure where the cost is not a function of time or cases. It also assumes the test cost and the misclassification cost have been defined on the same cost scale, such as the dollar cost incurred in a medical diagnosis. In the later part of this section, we will provide some details of this work because most of our work is based on this work.

Following the work in [10], further research has been done by us and our collaborators. Qin et al. [11] proposed a general framework for involving multiple costs in different cost scales. The task is to minimize one cost scale and control other cost scales in specified budgets. Chai et al. [13] proposed a test cost sensitive Naive Bayes network. Ling and Yang have done much work in test strategies in test cost sensitive learning [15, 16], and they aim to seek the best test attribute set for decision making. Zhang et al. [17] considers the cost sensitive learning in data with missing value and conclude that some data are left as unknown in domain of test cost sensitive learning and could be useful for decision.

Some representative strategies for test-cost-sensitive learning are briefly outlined as follows.

- **EG2**

EG2 [6] is a decision tree induction algorithm that uses the Information Cost Function (ICF) for selection of attributes. It is a modified version of ID3 (Quinlan 1989), the predecessor of a popular decision tree induction algorithm C4.5. ICF selects attributes based on both their information gain and their cost. The ICF for the *i*-th attribute, ICF*i*, is defined as follow:

$$ICFi = \frac{2^{Ii} - 1}{(Ci + 1)^{w}} \, . \tag{3}$$

Where *Ii* is the information gain associated with the *i*-th attribute at a given stage in the construction of the decision tree and *Ci* is the cost of measuring the *i*-th attribute, and *w* is an adjustable parameter between 0 and 1.

EG2 is able to reduce the overall cost by selecting attributes with less test cost and larger information gain to split.

- **CS-ID3**

CS-ID3 [7] is a decision tree algorithm that selects the split attribute which maximizes the following function:

$$\frac{Ii^2}{Ci}\,. \tag{4}$$

It is very similar to EG2. Where *Ii* is the information gain associated with the *i*-th attribute at a given stage in the construction of the decision tree and *Ci* is the cost of measuring the *i*-th attribute. However, CS-ID3 does not build a full decision tree then classify examples. Instead, it only constructs a lazy tree (a decision path) for each test example to classify them.

- **IDX**

IDX [5] is also a decision tree algorithm that selects the split attribute that maximizes the following function:

$$\frac{Ii}{Ci}\,. \tag{5}$$

Same as EG2 and CS-ID3, in the above function, *Ii* is the information gain associated with the *i*-th attribute at a given stage in the construction of the decision tree and *Ci* is the cost of measuring the *i*-th attribute. In C4.5, at each step, a greedy search strategy is used to choose the attribute with the highest information gain ratio. IDX uses a look-ahead strategy that looks n tests ahead, where n is a parameter that may be set by the user.

- **ICET**

ICET is a hybrid of a genetic algorithm and a decision tree induction algorithm. The genetic algorithm evolves a population of biases for the decision tree induction algorithm. The genetic algorithm is GENESIS [2]. The decision tree induction algorithm is EG2. ICET manipulates the bias of EG2 by adjusting the parameters C*i* and *w*. In the original design of EG2, C*i* is the attribute test cost. But in ICET, it is treated as a bias parameter.

In ICET, the genetic algorithm GENESIS begins with a population of 50 individuals. EG2 is run on each one of them to build a corresponding decision tree. The "fitness" of the individual is the total of test and misclassifications costs averaged over the number of cases. In the next generation, the population is replaced with new individuals generated from the previous generation. The fittest individuals in the first generation have the most offspring in the second generation. After a fixed number of generations, ICET stops and outputs the decision tree generated by the fittest individual.

- **MDP**

In Zubek and Dietterich's paper [8], cost-sensitive learning problem is cast as a Markov Decision Process (MDP), and solutions are given as searching in a state space for optimal policies. For a given new case, depending on the values obtained, the resulting policy can suggest an optimal action to perform in order to minimize both the misclassification and the test costs. Their admissible search heuristic is shown to reduce the problem search space remarkably. However, it may take very high computational cost to conduct the search process. In addition, to reduce over-fitting, they have introduced a supplementary pruning heuristic named statistical pruning.

- **Cost-sensitive Naive Bayes**

Chai et al. [13] presented a test Cost-sensitive Naive Bayes (CSNB) classifier which modifies the Naive Bayes classifier by including a test strategy which determines how unknown attributes are selected to perform test on in order to minimize the sum of misclassification cost and test cost. In the framework of CSNB, attributes are intelligently selected for testing to get both sequential test strategies and batch test strategies.

- **Cost-sensitive Decision Tree**

Cost-sensitive Decision Tree (CSDT) is based on C4.5. When building a decision tree, at each step, instead of choosing an attribute that minimizes the entropy (as in C4.5), CSDT chooses an attribute that reduces and minimizes the total of misclassification cost and test cost, for the split. Similar to C4.5, CSDT chooses a locally optimal attribute without backtracking. To make a decision tree cost-sensitive, the decision on which attribute to split on is determined by calculating the misclassification cost for every possible split, and, of course, choosing the lowest. Elkan (2001) points out that this approach may lead to a classification model that minimizes the cost of misclassification of the training set, but does not produce an optimal model when applied to unseen data, mainly because of over-fitting.

- **Multiple Scale Cost-sensitive Decision Tree**

Qin et al. [11] argue that cost sensitive decision tree algorithm must consider the resource budget when building trees and classifying test examples. Based on the decision tree built by Ling et al. [10], they propose a new decision tree algorithm which considers multiple cost scales in the tree building and testing process. In their algorithm, misclassification cost and test cost can be on the same scale if both of them can be converted to dollar values. They can be on different scales if one of them cannot be converted. Other resource costs such as time cost are always on different scales. When building decision trees, instead of using total cost as the split criterion, they use cost gain ratio (cost gain / resource cost) to split. In their paper, a resource budget is set for each resource. During the testing, if an example is run out of resource, it has to stop at the internal node which represents the attribute. Missing values are handled in the same way as that in Ling et al. [10].

The aim of Qin et al.'s decision tree algorithm [10] is to minimize the misclassification cost with limited resource budget. In their framework, misclassification cost does not have a budget. All other resources have limited budget. Multiple resource costs, such as test cost, time cost and computation cost, can be involved in the decision process. The decision tree built by Ling et al. (2004) becomes a special case of this more general cost sensitive decision tree building framework.

3.2 Lazy Cost-Sensitive Learning with Medical History

When dealing with complex and versatile medical data, the different nature of individual attributes of the data may require different measures and usage requirements, i.e. an attribute value can be measured at costs distinguished from other attributes and this value is with its specific usage period. Below Example 1 shows an actual and new setting of cost-sensitive learning.

Example 1: Assume that the cost of testing X needs 1 day, $3000 and 100ml blood; and Y 5 days, $1000 and 50ml blood. The three costs (testing time, testing fee and blood need) are valued in distinct scales. If a patient is too weak to provide more than 60ml blood, then his/her doctor can only choose test Y for him; or if the patient needs an urgent decision within 3 days, then test X should be much more proper for him. On the other hand, if the patient holds a valid test result of X in his/her medical history, the doctor can certainly reset X's test cost to zero before considering further tests.

This setting faced by cost-sensitive learning has two new features: the multiple-scale cost constraint and combination of test data with medical history. Unfortunately, existent cost sensitive learning methods do not handle well the above complicate but real problem because they often simplify the costs in a uniform scale, and in particular, they are not designed for those cases that patients are with certain history record.

Our research attacks this new setting of cost-sensitive learning with a new cost model based on existing cost-sensitive learning framework. This model is introduced with some new concepts, such as target cost and resource cost, and a multiple-scale cost structure which represents the interrelationship between the new cost concepts and the multiple costs involved during a learning process. With the new cost structure, an attribute selection strategy is incorporated to a lazy decision tree induction, so as to minimize the total cost of multiple-scale costs when medical history is dynamically utilized to current test tasks.

3.3 Semi-cost-Sensitive Learning

Another research my research group is to apply cost-sensitive learning techniques to semi-supervised environments.

In many real world applications, labeled examples are often very difficult, time consuming or expensive to obtain, as they require the efforts of human annotators. At the mean time, unlabeled data sometimes is easy to get, but they need to be used carefully. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build classifiers with higher accuracy and less cost [14]. However, semi-supervised classification still needs to tackle he unbalance class distribution problem and it is not well studied. We apply semi-supervised learning techniques to learn cost-sensitive models from datasets with inadequate labeled data.

We propose two classification strategies for learning cost-sensitive classifier from training datasets with both labeled and unlabeled data, based on Expectation Maximization (EM). The first method, Direct-EM uses EM to build a semi-supervised classifier, then directly compute the optimal class label for each test example using the class probability produced by the learning model. The second method, CS-EM modifies EM by incorporating misclassification cost into the probability estimation process.

4 Conclusions

We have introduced the up-to-date prevailing cost-sensitive learning methods. Some open research topics include

- Semi-learning strategies for cost-sensitive classifiers;
- Instance-based (Lazy) learning strategies for cost-sensitive classifiers;

- Semi-lazy learning strategies for cost-sensitive classifiers;
- Cold deck learning strategies for cost-sensitive classifiers;
- Cost-sensitive learning from multiple data sources.

Acknowledgement. This work was supported in part by the Australian Research Council (ARC) under grant DP0985456, the Nature Science Foundation (NSF) of China under grant 90718020, and the Guangxi NSF (Key) grants.

References

1. Turney, P.: Types of cost in inductive concept learning. In: Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning, pp. 15–11. Stanford University, California (2000)
2. Turney, P.D.: Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of Artificial Intelligence Research 2, 369–409 (1995)
3. Elkan, C.: The foundations of cost-sensitive learning. In: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, pp. 973–978. Morgan Kaufmann Publishers Inc., Seattle (2001)
4. Margineantu, D.D., Dietterich, T.G.: Improved class probability estimates from decision tree models. Lecture Notes in Statistics, vol. 171, pp. 169–184. Springer, New York (2002)
5. Norton, S.W.: Generating better decision trees. In: Proceedings of the Eleventh International Conference on Artificial Intelligence, pp. 800–805. Morgan Kaufmann Publishers Inc., Detroit (1989)
6. Núñez, M.: The use of background knowledge in decision tree induction. Machine Learning 6(3), 231–250 (1991)
7. Tan, M.: Cost-sensitive learning of classification knowledge and its applications in robotics. Machine Learning 13(1), 7–33 (1993)
8. Zubek, V.B., Dietterich, T.G.: Pruning Improves Heuristic Search for Cost-Sensitive Learning. In: Proceedings of the Nineteenth International Conference on Machine Learning, pp. 27–34. Morgan Kaufmann Publishers Inc., San Francisco (2002)
9. Greiner, R., Grove, A.J., Roth, D.: Learning cost-sensitive active classifiers. Artificial Intelligence 139(2), 137–174 (2002)
10. Ling, C.X., Yang, Q., Wang, J., Zhang, S.: Decision trees with minimal costs. In: ICML 2004, p. 69. ACM, Banff (2004)
11. Qin, Z., Zhang, C., Zhang, S.: Cost-sensitive Decision Trees with Multiple Cost Scales. In: Webb, G.I., Yu, X. (eds.) AI 2004. LNCS (LNAI), vol. 3339, pp. 380–390. Springer, Heidelberg (2004)
12. Wang, T., Qin, Z., Zhang, S.: Cost-sensitive Learning - A Survey. Accepted by International Journal of Data Warehousing and Mining (2010)
13. Chai, X., Deng, L., Yang, Q., Ling, C.X.: Test-Cost Sensitive Naive Bayes Classification. In: ICDM 2004, pp. 51–58. IEEE Computer Society Press, Brighton (2004)
14. Zhu, X., Wu, X.: Cost-Constrained Data Acquisition for Intelligent Data Preparation. IEEE Transactions on Knowledge and Data Engineering 17(11), 1542–1556 (2005)
15. Sheng, S., Ling, C.X., Yang, Q.: Simple Test Strategies for Cost-Sensitive Decision Trees. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 365–376. Springer, Heidelberg (2005)

16. Sheng, V.S., Ling, C.X., Ni, A., Zhang, S.: Cost-Sensitive Test Strategies. In: Proceedings of 21st National Conference on Artificial Intelligence (AAAI 2006), pp. 482–487. AAAI Press, Boston (2006)
17. Zhang, S., Qin, Z., Ling, C.X., Sheng, S.: Missing Is Useful: Missing Values in Cost-Sensitive Decision Trees. IEEE Transactions on Knowledge and Data Engineering 17(12), 1689–1693 (2005)
18. Qin, Z., Zhang, S., Liu, L., Wang, T.: Cost-sensitive Semi-supervised Classification using CS-EM. In: IEEE 8th International Conference on Computer and Information Technology, pp. 131–136. IEEE Computer Society Press, Sydney (2008)